4 COMPUTER PROGRAMMING

JAVA OPERATORS

CONTENTS

- Operators and Expression
- Arithmetic / Relational / Logical operators
- Utilizing examples
- String

Operators and Expressions

Types of operators

- Unary operator
  - Prefix operator and the suffix (postfix) as an operator
- Binary operator
  - **op1 operator op2**
- Ternary operator

Arithmetic operators

operators	usage	explanation	remarks
+	op1+op2	op1과 op2를 더한다	단항 및 이항
–	op1–op2	op1과 op2를 뺀다	단항 및 이항
*	op1*op2	op1과 op2를 곱한다	이항
/	op1/op2	Op1을 op2로 나눈다	이항
%	op1%op2	Op1을 op2로 나눈 나머지를 구한다	이항
++	op++	op값 1 증가, op값을 증가시키기 전에 평가	단항
	++op	op값 1 증가, op값을 증가시킨 다음평가	단항
––	op––	op값 1 감소, op값을 감소시키기 전에 평가	단항
	––op	op값 1 감소, op값을 증가시킨 다음평가	단항

Relational operators

- A binary operator
- Compare the values of two operands and return a true / false as a result

Logical operators

- To assess the value of the operand returns a true / false value.

Bitwise operators

operators	usage	explanation
>>	op1 >> op2	op1을 op2만큼 오른쪽으로 쉬프트(shift)
<<	op1 << op2	op1을 op2만큼 왼쪽으로 쉬프트(shift)
>>>	op1 >>> op2	op1을 op2만큼 오른쪽으로 쉬프트하면서 왼쪽에는 항상 부호에 무관하게 0으로 채워짐
&	op1 & op2	비트 단위의 논리곱(AND)
|	op1 | op2	비트 단위의 논리합(OR)
^	op1 ^ op2	비트 단위의 배타적 논리합(XOR)
~	~op	비트 단위의 보수

< 비트 연산자의 논리표 >

Assignment operators

operators	usage	meaning
+=	op1 += op2	op1 = op1 + op2
-=	op1 - = op2	op1 = op1 – op2
*=	op1 * = op2	op1 = op1 * op2
/=	op1 / = op2	op1 = op1 / op2
%=	op1 %= op2	op1 = op1 % op2
&=	op1 &= op2	op1 = op1 & op2
|=	op1 |= op2	op1 = op1 | op2
^=	op1 ^= op2	op1 = op1 ^ op2
<<=	op1 <<= op2	op1 = op1 << op2
>>=	op1 >>= op2	op1 = op1 >> op2
>>>=	op1 >>>= op2	op1 = op1 >>> op2

Ternary operators

- “ ? : ”
- Operator to be used to abbreviate a if-then-else selection statement

Expression1 ? Expression2 : Expression3

- example
  - `flag = count > 0 ? 0 : 1 ;`

Operator precedence

Ex1 for operators and expressions

```
 1 public class Arithmetic {
 2   public static void main(String args[]) {
 3     int a=5, b=2 ;
 4     int sum=a+b;
 5     int sub=a-b;
 6     int mul=a*b;
 7     float div=a/b;
 8     int mod=a%b;
 9     a++;
10     b--;
11
12     System.out.println("a+b=" + sum);
13     System.out.println("a-b=" + sub);
14     System.out.println("a*b=" + mul);
15     System.out.println("a/b=" + div);
16     System.out.println("a%b=" + mod);
17     System.out.println("a의 단항증가연산="+a);
18     System.out.println("b의 단항감소연산="+b);
19     }
20 }
```

Ex2 for operators and expressions

```
 1 class Bitwise {
 2   public static void main(String args[]) {
 3     int a = 2;
 4     int b = 5;
 5     int c = a | b;
 6     int d = a & b;
 7     int e = a ^ b;
 8     int i;
 9     int j;
10     i = a << 2;
11     j = b >> 2;
12
13     System.out.println("        a = " + a);
14     System.out.println("        b = " + b);
15     System.out.println("      a|b = " + c);
16     System.out.println("      a&b = " + d);
17     System.out.println("      a^b = " + e);
18     System.out.println("     a<<2 = " + i);
19     System.out.println("     b>>2 = " + j);
20   }
21 }
```

Ex3 for operators and expressions

```
1  class BitEquals {
2    public static void main(String args[]) {
3      int a = 10;
4      int b1 = 5, b2 = 5, b3 = 5;
5
6      System.out.println("a = " + a);
7      a += 4;
8      System.out.println("a += 4 의 결과 " + a);
9      a -= 4;
10     System.out.println("a -= 4 의 결과 " + a);
11     a *= 4;
12     System.out.println("a *= 4 의 결과 " + a);
13     a /= 4;
14     System.out.println("a /= 4 의 결과 " + a);
15     a %= 4;
16     System.out.println("a %= 4 의 결과 " + a);
17     a |= 4;
18     System.out.println("a |= 4 의 결과 " + a);
19
20     System.out.println("b1 = " + b1);
21     System.out.println("b2 = " + b2);
22     System.out.println("b3 = " + b3);
23     b1 >>= 1;
24     b2 <<= 1;
25     b3 >>>= 1;
26     System.out.println("b1 >>= 1 의 결과 " + b1);
27     System.out.println("b2 <<= 1 의 결과 " + b2);
28     System.out.println("b3 >>>= 1 의 결과 " + b3);
29   }
30 }
```

Ex4 for operators and expressions

```
1 class TernaryDemo {
2   public static void main(String args[]) {
3     int i=10;
4     System.out.print("정수형 변수 i의 값은 " + i + " 이며 ");
5     String str = (i % 2 == 0) ? "짝수 " : "홀수 ";
6     System.out.print(str);
7     System.out.println("입니다");
8   }
9 }
```

STRING

- Java implements strings as objects of the String class
- Example
  - print string
  - print a string combining

CONCLUDE

- Operators and Expression
- Arithmetic / Relational / Logical operators
- Utilizing examples
- String